Amendments to the Drawings:

The attached sheets of drawings include changes to Figs 4B and 5. Sheet 6, which includes Fig. 4B, replaces the original sheet including Fig. 4B. In Fig. 4B, is amended to replace the reference "M" with --MD--. Sheet 8, which includes Fig. 5, replaced the original sheet 8 including Fig. 5. In Fig. 5, the word "on" is replaced with --m-- in the last line in Step 51; two occurrences of "S1" are replaced with --S3-- in Step 52; the term "S2" is replaced with --S4-- in Step 52; the term "M2" is replaced with --MD2-- in Step 55; and replace "2nd" with --3rd-- in Step 57.

Attachment: Replacement Sheets

REMARKS/ARGUMENTS

In this amendment, the drawings, Fig. 4B and Fig. 5, and the specification, paragraphs [0005], [0051] and [0052], are amended to correct literal errors and remedy improper expressions. No new matter is introduced.

In this amendment, claims 1, 3-6, 9, 11-12, 14, 16 and 18 are amended; claims 2 and 10 are cancelled; and new claims 21 and 22 are added. Support is found from the original specification. No new matter is introduced. In order to further clarify the support of currently pending claims, numeral references corresponding to features in the drawings are given bolded as follows for illustration purpose only.

1. (Currently Amended) A data access method, comprising a data reading procedure to read a data **(data 3-12, Fig. 4A)** from a data storage zone **(20)** wherein said data is stored in a bit range of said data storage zone **(20)** covering at least one storage unit **(B21, B22)**, each storage unit **(B21, B22)** of said data storage zone **(20)** consisting of m bits, and said bit range consists of n bits from a starting bit address (a) **(3)** to an end bit address (b) **(12)**, and said data reading procedure comprising steps of:

   i) performing a first operation **(mod[3,8])** of said starting bit address (a) **(3)** to obtain a first shift S1 **(3)**;

   ii) performing a second operation **(8-mod[3,8])** of said starting bit address (a) **(3)** to obtain a second shift S2 **(5)**;

   iii) performing a first shift operation **(R1)** of said data **(data 3-12)** with said first shift S1 **(3)** to obtain a first shifted data unit **(T21)**;

   iv) performing a second shift operation **(L1)** of said data **(data 3-12)** with said second shift S2 **(5)** to obtain a second shifted data unit **(T22)**;

   v) synthesizing said first and said second shifted data units **(T21, T22)** to obtain a read data unit **(R21)**; and

   vi) repeating at least one of said steps iii), iv) and v) when n is greater than m.

2. (Cancelled).

3. (Currently Amended) The data access method according to claim 1 wherein said first and said second operations are performed by the following formulae:
   S1 = mod [a, m]; and
   S2 = m – mod [a, m] = m – S1,
   where mod [a, m] is the remainder on division of a by m.

4. (Currently Amended) The data access method according to claim 3 wherein said first shift operation **(R1)** is performed by shifting a first portion of said data **(data 3-7)** stored in a first storage unit **(B21)** of said data storage zone **(20)** toward one of the

higher bit direction and the lower bit direction **(the lower bit direction)**, and said second shift operation **(L1)** is performed by shifting a second portion of said data **(data 8-12)** stored in a second storage unit **(B22)** of said data storage zone **(20)** toward the other of the higher bit direction and the lower bit direction **(the higher bit direction)**.

5. (Currently Amended) The data access method according to claim 4 wherein said second storage unit **(B22)** is immediately adjacent to said first storage unit **(B21)** in said data storage zone **(20)**.

6. (Currently Amended) The data access method according to claim 5 wherein in said step (vi), only said step (iii) is repeated for shifting an end data unit **(data 8-15)** comprising said end data bit address (b) **(12)** with said first shift S1 to obtain a last shifted data unit **(T23)**.

7. (Original) The data access method according to claim 6 further comprising a step of masking said last shifted data unit **(T23)** with a mask data MD for clearing bits **(10-15)** excluded from said bit range, where MD = 0xFF >> (m - (b-a+1)), the expression "0xFF" indicates an 8-bit hexadecimal mask data and the 8 bits are all "1", and the expression "X >> Y" indicates the rightward shift of the data X by Y bits.

8. (Original) The data access method according to claim 1 wherein said first and said second shifted data units **(T21, T22)** are synthesized via an OR gate operation **(O1)**.

9. (Currently Amended) A data access method, comprising a data writing procedure to write a data **(data 3-12, Fig. 6A, 6B)** into a data storage zone **(30)**, said data storage zone **(30)** storing said data in a bit range **(bits 3-12)** covering at least one storage unit **(B31, B32)**, each storage unit **(B31, B32)** of said data storage zone **(30)** consisting of m bits, said bit range **(bits 3-12)** consisting of n bits from a starting bit address (a) **(3)** to an end bit address (b) **(12)**, and said data writing procedure comprising steps of:

i) performing a first operation **(mod [3,8])** of said starting bit address (a) **(3)** to obtain a first shift S3 **(3)**;

ii) performing a second operation **(m-mod [3,8])** of said starting bit address (a) **(3)** to obtain a second shift S4 **(5)**;

iii) performing a first shift operation **(R2, Fig. 6B)** of said data with said second shift S4 **(5)** to obtain a first shifted data unit **(T32)**;

iv) performing a second shift operation **(L2)** of said data with said first shift S3 **(3)** to obtain a second shifted data unit **(T33)**; and

v) synthesizing said first and said second shifted data units **(T32, T33)** to obtain a written data unit **(T34)**,

wherein at least one of said steps iii), iv) and v) is preformed more than once when n is greater than m.

10. (Cancelled)

11. (Currently Amended) The data access method according to claim 9 wherein said first and said second operations are performed by the following formulae:
S3 = mod [a, m]; and
S4 = m – mod [a, m] = m – S3,
where mod [a, m] is the remainder on division of a by m.

12. (Currently Amended) The data access method according to claim 11 wherein said first shift operation **(R2)** is performed by shifting a first data unit **(data 3-10)** of said data to be written toward one of the higher bit direction and the lower bit direction **(the lower bit direction)** to obtain said first shifted data unit **(T32)**, and said second shift operation **(L2)** is performed by shifting a second data unit **(data 11-12)** of said data to be written toward the other of the higher bit direction and the lower bit direction **(the higher bit direction)** to obtain said second shifted data unit **(T33)**.

13. (Original) The data access method according to claim 12 wherein said second data unit **(data 11-12)** is immediately adjacent to said first data unit **(data 3-10)** in said data storage zone **(30)**.

14. (Currently Amended) The data access method according to claim 13 further comprising steps of:
determining whether said second data unit **(data 11-12)** is the last data unit of said data to be written, wherein said first shifted data unit **(T32)** and said second shifted data unit **(T33)** are synthesized to obtain an end written data unit **(T34)** when said second data unit **(T33)** is the last data unit of said data to be written; and
performing a masking procedure with a mask data MD3 for clearing bits **(bits 8-12)** of an end storage unit **(B32)** of said storage zone **(30)** for storing said end written data unit **(T34)** when said second data unit **(T33)** is the last data unit of said data to be written, where MD3 = 0xFF << (mod [b, m] +1), mod [b, m] is the remainder on division of b by m, the expression "0xFF" indicates an 8-bit hexadecimal mask data and the 8 bits are all "1", and the expression "X << Y" indicates the leftward shift of the data X by Y bits.

15. (Original) The data access method according to claim 13 wherein said first and said second shift operations **(R2, L2)** are further performed on subsequent data units until the last data unit of said data to be written has been shifted.

16. (Currently Amended) The data access method according to claim 12 further comprising before said step (iii) steps of:
determining whether said first data unit **(data 3-10)** is the starting data unit of said data to be written;
performing a starting shifting operation **(L2)** of said first data unit **(data 3-10)** with said first shift S3 **(3)** to obtain a starting shifted data unit **(T31)** when said first data unit **(data 3-10)** is the starting data unit of said data to be written; and

performing a masking procedure with a mask data MD2 for clearing bits **(bits 3-7)** of a starting storage unit **(B31)** of said storage zone **(30)** for storing said starting shifted data unit,

where MD2 = ~(0xFF << S3), the expression "0xFF" indicates an 8-bit hexadecimal mask data and the 8 bits are all "1", the expression "X << Y" indicates the leftward shift of the data X by Y bits, and the expression "~Z" indicates the reverse logic operation of data Z.

17. (Original) The data access method according to claim 9 wherein said first and said second shifted data units **(T32, T33)** are synthesized via an OR gate operation **(O2)**.

18. (Currently Amended) A data access method, comprising a data writing procedure to write a data **(data 3-12, Fig. 6A, 6B)** into a data storage zone **(30)**, said data storage zone **(30)** storing data in a bit range **(bits 3-12)** covering at least one storage unit **(B31, B32)**, each storage unit **(B31, B32)** of said data storage zone **(30)** consisting of m bits, said bit range **(bits 3-12)** consisting of n bits from a starting bit address (a) **(3)** to an end bit address (b) **(12)**, and said data writing procedure comprising steps of:

performing a first clear and writing procedure of said data to be written when n is no greater than m **(Step 54, Fig. 5)**, said first clear and writing procedure comprising a step of masking said bit range with a first mask data MD1 = ~((0xFF >> ((m-1) – b + a)) << mod [a, m]); and

performing a second clear and writing procedure and a third clear and writing procedure of said data to be written when n is greater than m **(Steps 55-58)**, said second clear and writing procedure comprising a step of masking a starting storage unit with a second mask data MD2 = ~(0xFF << mod [a, m]), and said third clear and writing procedure comprising a step of masking an end storage unit with a third mask data MD3 = 0xFF << (mod [b, m] +1);

where the expression "0xFF" indicates a hexadecimal mask data, the expression "X >> Y" indicates the rightward shift of the data X by Y bits, the expression "X << Y" indicates the leftward shift of the data X by Y bits, the expression "~Z" indicates the reverse logic operation of data Z, the expression "X & Y" indicates AND gate operation of data X and Y, the expression "mod [a, m]" indicates the remainder on division of a by m, and the expression "mod [b, m]" indicates the remainder on division of b by m.

19. (Original) The data access method according to claim 18 wherein said data writing procedure is performed as little endian.

20. (Original) The data access method according to claim 18 wherein said data writing procedure is performed as big endian.

21. (New) The data access method according to claim 18 wherein when n is greater than m, the starting data unit of said data is shifted by a shift S3 and then written into said starting storage unit of said data storage zone in said second clear and writing

procedure, where S3 = mod [a, m] that is the remainder on division of a by m **(Step 55)**.

22. (New) The data access method according to claim 18 wherein when n is greater than m, the last second data unit and the last data unit of said data are shifted by a first shift S3 and a second shift S4, respectively, and the differentially shifted data are synthesized and then written into said end storage unit in said third clear and writing procedure, where S3 = mod [a, m] that is the remainder on division of a by m, and S4 = m – S3 **(Step 57)**.

Claim Rejections – 35 USC § 102 & 103.

In the Office Action, the Examiner rejected claims 1-6, 8-13, 15 and 17 under 35 USC 102(e) as being anticipated by Roussel et al. (US Patent No. 6,721,866; hereinafter Roussel), rejected claims 7, 13, 16 and 18 under 35 USC 103(a) as being unpatentable over Roussel in view of Debes et al. (US Patent Application No. 2003/0084082); hereinafter Debes), and further rejected claims 19 and 20 under 35 USC 103(a) as being unpatentable over Roussel and Debes further in view of Hensen et al. (US Patent No. 4,814,976). The Applicant respectfully traverses the rejections for the following.

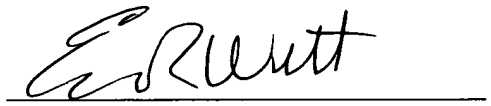
The Applicant submits that Roussel does not teach or suggest all elementary steps of the invention. For example, Roussel does not teach or suggest repeating one or more of the steps when n is greater than m. According to the amended independent claims 1 and 9 of the present application, the data access method is applicable to the condition that the bit number extends over one data storage unit, *i.e.* n is greater than m, by repeating one or more of the steps. On the contrary, Roussel's Fig. 4B and Fig. 5 cited by the Examiner only teach the shift of first and second operands and the combination of the shifted operands. Accordingly, the invention is not anticipated by Roussel.

It is understood from the abstract, Roussel's invention is to obtain an aligned operand. An operand, according to Roussel, is a fundamental data type in memory, whose size are bytes, words, doublewords, quadwords, and double quadwords (col. 3, lines 34-38). In other words, no inconsistent size between the data storage unit and the bit range to be accessed need be dealt with by Roussel. Since there is teaching or suggestion in Roussel how to access data within a bit range inconsistent with the data storage unit, the invention would not have been obvious over

Roussel. In addition, Roussel in view of Debes or further in view of Hensen are silent in this aspect.

For the above reasons, Applicant respectfully submits that claims 1, 3-9 and 11-22 are neither anticipated nor obvious from the cited prior art references. Applicant respectfully requests that a timely Notice of Allowance be issued in this case. If there are any remaining issues preventing allowance of the pending claims that may be clarified by telephone, the Examiner is requested to call the undersigned.

Respectfully submitted,



Evan R. Witt
Reg. No. 32,512
Attorney for Applicant

Date: May 15, 2006

MADSON & AUSTIN
Gateway Tower West
15 West South Temple, Suite 900
Salt Lake City, Utah 84101
Telephone: 801/537-1700